

**AGENDA FOR EXAMINER GARY J. PORTKA INTERVIEW**

**Application No.: 11/753,608 [112056-0148]**

**Date: 01/04/08 3:00 PM**

Applicant thanks the Examiner for the upcoming interview. Below are the detailed topics and arguments which the Applicant's agent intends to make during the interview. Please feel free to contact Steve LeBarron, Reg. No. 62, 479 at any time prior to the interview with any questions you might have. Agent's direct line is 617-951-3016.

**Rejection Under 35 U.S.C. §102(b)**

At paragraph 7 of the Office Action, claims 1-14 and 16-72 were rejected under 35 U.S.C. §102(b) as being unpatentable in view of Permut U.S. Patent No. 6,260,115, issued on July 10, 2001 (hereinafter "Permut").

Applicant respectfully urges that Permut has no disclosure of Applicant's claimed novel *adjusting an amount of readahead data to retrieve from the data container, based on the plurality of factors stored within the readset data structure.*

Applicant's claimed invention is a method and system for a storage operating system, implemented in a storage system, to optimize an amount of readahead data retrieved from a data container of the storage system. In particular, a plurality of readset data structures are **maintained for a selected file of the plurality of files. Each of the readset data structures hold a plurality of factors for a selected readstream that allow the system to adaptively adjust the readahead data** accordingly. The system initiates when a client read request is **received at the storage system for a particular read request.** Once the request has been received, the system locates a readset data structure for the particular read stream so that the system can determine whether the storage operating system is permitted to retrieve readahead

data from the data container in response to the received client read request. If it is determined that the storage operating system is permitted to retrieve readahead data from the data container, the system acts accordingly and (i) *adjust the amount of readahead data to retrieve from the data container. This adjustment is based on the plurality of factors stored within the readset data structure, (ii) retrieves the adjusted amount of readahead data from the data container, and (iii) determines if the readset data structure meets a criteria for being updated, and if the readset data structure meets the criteria, updating the readset data structure.*

Permut does not disclose adjusting the amount of readahead data to retrieve from the data container based on a plurality of factors. Permut is merely a flag based system which indicates whether the system should or should not retrieve a predetermined, fixed amount of data using Permut's prestaging process. This type of system is like that of the prior art discussed on page 5 of the Applicant's background. Furthermore, as discussed above, Permut does not use a plurality of factors to adjust the amount of data that is retrieved from the data container. Permut assigns a predetermined fixed amount (e.g., hints) connected to file for the system to use the next time the data is retrieved. In Applicant's retrieval process, the amount of data collected during the readahead step is determined by the system each time there is a new read request. **This amount is determined based on a plurality of current system factors not on a previously established fixed rate like in Permut.**

#### **Applicant's Response to Examiner's Response to Arguments**

Applicant respectfully points out there is no mention of "*adjusting* an amount of readahead data *to retrieve from the data container*, based on the plurality of factors stored within the readset data structure." Permut tells the system to retrieve a fixed amount of previously chosen data every time the system retrieves data for a particular file. Permut does not adjust the data to be retrieved every time the data is retrieved. By definition, *adjust* means changing. Therefore, a fixed amount can not be changed (i.e., adjusted). The definition provided above is supported in Applicant's Specification on page 21, Lines 16-23, which state:

"The level value 604 may be *adjusted ... after every client read request is processed by the file system 260, until the readset 600 is deallocated or reused, as described below*. This aging process may be subject to various conditions. For

example, if the level value 604 is decremented to a value that is less than the predetermined initial value (e.g., equal to 10), then the next time the level value is incremented, the level value may be set equal to the predetermined initial level value. Further, the file system 260 may ensure that the level value 604 is not incremented past its predetermined upper-bound value nor decremented below its predetermined lower-bound value.”

Further page 29, Lines 2–10 state

“Specifically, the total number of blocks to read 714 is *adjusted* to include both the client-requested must-read data blocks 716 as well as the number of speculative readahead data blocks 718 determined by the steps 920-945. Here, it is noted that the file system 260 searches the buffer cache 156 to determine whether any of the requested data blocks are already stored in in-core buffers 1200. If so, the file system appropriately adjusts the disk I/O hint 710 to retrieve only those data blocks not already accessible in the buffer cache. Moreover, the file system may have to generate more than one disk I/O hint 710 if more than one non-contiguous sequence of file data blocks (i.e., fbns) are requested from disk storage.”